

BYTESAVER

INSTRUCTION MANUAL

- **Bytesaver Assembly**
- **Bytesaver Parts List**
- **Bytemover Software**
- **2708-2704 PROM Data**
- **Bytemover Assembly Listing**

© Copyright 1977. All rights reserved.

Introduction

The Cromemco Bytesaver is a read/write, non-volatile memory board, plug compatible with the Standard-100 (S-100) microcomputer bus. The Bytesaver has the capacity for eight 2708 U.V. erasable PROMs for a full 8K bytes of memory.

The Bytesaver contains an integral PROM programmer along with a DC-to-DC supply for generating the programming voltage. Programming is accomplished by a series of memory write operations to the PROM being programmed.

Cromemco provides the necessary programming software. Our Bytemover software, described later in this manual, allows convenient PROM programming using your computer's front panel sense switches to control the operation (e.g. to select one of the eight PROMs to be programmed). Programming can also be carried out using the Z-80 Monitor supplied with our Z-80 CPU card and our Z-80 microcomputer system.

Assembly Instructions

The Cromemco Bytesaver™ kit can be assembled in about one evening. All components are mounted on the component side of the p.c. board (the side with the printed legend) and soldered on the opposite side. Be sure to use high-quality, rosin core solder for the assembly and a fine-tipped, low-wattage soldering iron.

1. Solder the 10 14-pin IC sockets, the 6 16-pin IC sockets and the 8 24-pin sockets in position.

2. Solder the following ¼ Watt resistors in position:

✓ R1	47K	yellow-violet-orange
✓ R2	10K	brown-black-orange
✓ R3	180	brown-gray-brown
✓ R4	1K	brown-black-red
✓ R5	9.1K	white-brown-red
✓ R6	1.5K	brown-green-red
✓ R7	1.2K	brown-red-red
✓ R8	47	yellow-violet-black
✓ R9	1K	brown-black-red
✓ R10	10	brown-black-black
✓ R11	5.6K	green-blue-red
✓ R12	5.6K	green-blue-red
✓ R13	10K	brown-black-orange
✓ R14	5.6K	green-blue-red
✓ R15	180	brown-gray-brown
✓ R16-R31	18K	brown-gray-orange
✓ R32-R39	4.7K	yellow-violet-red

3. Next, install the 1N914 diodes. **We recommend that no diode be installed in the diode position just below transistor Q0.** Since we recommend that the PROM containing the Bytemover software be inserted in PROM position zero, installing this diode may allow accidental programming of this PROM.

When installing the diodes, be careful to orient them properly, by noting the position of the cathode (banded) end. Due to the close spacing of the holes in the p.c. board, all diodes should be mounted on end.

✓ 4. Install the 23 capacitors as shown on the p.c. board. Be careful to orient the electrolytic capacitors with the positive (+) end as shown on the board.

✓ 5. Solder the transistors in place taking care to orient them properly. Note that Q8 and Q9 are 2N3906 transistors and Q10 is a type MPS6560. All other transistors are type 2N3904.

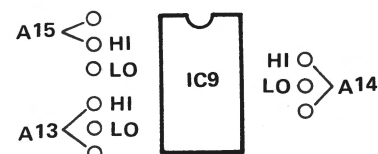
✓ 6. Install the p.c. board switch (SW1) in the upper left corner of the p.c. board.

✓ 7. Install the Cromemco high-speed pulse transformer (Model XT8K) in position T1. Note that the leads are asymmetrically arranged so there is only one possible orientation.

✓ 8. Install IC14, the positive 12V regulator IC. Use a 6-32 by 1/4" screw and nut.

✓ 9. Initially install the heatsink in the upper right corner of the p.c. board by just starting the nuts on the 6-32 by 3/8" screws. Install IC12 and IC13 but be sure to place the insulating washer between IC13 and the heatsink. **The nylon screw must be used to secure IC13.** It is important that the screw be inserted from the p.c. board side so the screw head is against the foil side. Be aware that the insulating washer may have to be trimmed with a pair of scissors to clear the protrusions in the heatsink. Tighten the nuts on the screws in the heatsink assembly only after all the screws have been inserted. Take care that the leads of the voltage regulators do not come in contact with the sides of the openings in the heatsink. **Although voltage regulators IC12, IC13 and IC14 may look similar, they are not interchangeable.**

10. Install three jumper wires to select where the Bytesaver is to reside in memory. Each of the three high-order address lines (A15, A14 and A13) may be tied to either the corresponding "H" or "L" terminal. For example, in order for the Bytesaver to reside in the top 8K of memory, the three jumpers should be installed as shown below:



This adjustment causes the Bytesaver to reside in the top 8K of the memory map.

11. Install the ICs in their sockets, being careful to orient pin 1 of each IC as shown by the small white dot on the p.c. board at each IC position. Install a PROM containing the Bytemover software in PROM position zero.

Your Bytesaver is now fully assembled. Detailed operating instructions are given in the Bytemover software section of this manual.

Notes

Interrupts: If you plan to have your computer respond to interrupts while executing a program stored in the Bytesaver memory, a small modification is required to the Bytesaver circuit. This modification is shown in Note 1 on the Bytesaver schematic.

Wait State: Should you wish to use low speed 2704s or 2708s (access times greater than 450 ns) in your Bytesaver, be aware there is a provision for a wait state. Simply insert the jumper wire between IC10 and IC11. No jumper need be inserted when using full speed PROMs. Jumper is also required for 4MHz operation.

Programming PROMs

PROM programming with the Bytesaver is carried out by a sequence of memory write operations to the Bytesaver card. A sequence of approximately 100 memory write operations is required to write the data into each location of every PROM that you wish to program. This sequence of memory write operations is carried out automatically when using the special instructions of Cromemco software. The Cromemco Z-80

Monitor, the Resident Operating System, and the 3K Control Basic Interpreter all have provision for PROM programming with the Bytesaver card. If your computer is equipped with front panel sense switches, our Bytemover software can be used for PROM programming. Operation of the Bytemover software is described in the next section.

Bytesaver Parts List

Capacitors		Integrated Circuits	
C1-C8	.01 μ F	IC1	.74123
C9-C15	10 μ F, 50V	IC2	.7474
C16	.005	IC3	.7402
C17	680 pF	IC4	.7406
C18	.01 μ F	IC5	.7406
C19	680 pF	IC6	.7402
C20	150 pF	IC7	.7406
C21-C23	.01 μ F	IC8	.7442
Diodes		IC9	.74L04
D1-D19	1N914 or 1N4531	IC10	.7410
Transistors		IC11	.74LS04
Q0-Q7	2N3904	IC12*	340T-5.0 or 7805
Q8, Q9	2N3906	IC13*	320T-5.0 or 7905
Q10	MPS6560	IC14*	340T-12 or 7812
Q11, Q12	2N3904	IC15	.7432 or 74LS32
Resistors		IC16	.74367
R1	47K	IC17	.74367
R2	10K	IC18	.74367
R3	180	IC19	.74367
R4	1K	Miscellaneous	
R5	9.1K	SW1	.p.c. board switch
R6	1.5K	T1	XT8K pulse transformer
R7	1.2K	Sockets	
R8	.47	10	14-pin
R9	1K	6	16-pin
R10	10	8	24-pin
R11	5.6K	Hardware	
R12	5.6K	3	.6-32 by 3/8" screws
R13	10K	1	.6-32 by 1/4" screw
R14	5.6K	1**	.6-32 by 3/8" nylon screw
R15	180	5	6-32 nuts
R16-R31	18K	1	Heatsink
R32-39	4.7K	1	Insulating washer

Notes

*The three voltage regulator ICs (IC12, IC13 & IC14) may look physically similar, however they are not interchangeable. Each must be in the proper IC location.

**The nylon screw is used to secure IC13. It is important that the screw be inserted from the p.c. board side of the assembly so the head of the screw is against the foil side of the board.

Introduction

Cromemco Bytemover software is designed to be used with the Cromemco 8K Bytesaver described. When you purchase a Bytesaver with one 2708 PROM, the Bytemover software is preprogrammed in that PROM.

The PROM containing the Bytemover software is normally inserted into PROM location zero on the Bytesaver board.

The Bytemover software can be used to program a PROM in any of the PROM locations on the Bytesaver board. The Bytemover software can also be used to transfer programs from PROM to RAM.

The operation of the Bytemover software is controlled by setting front panel sense switches on any S-100 bus-compatible computer. However, to use the Bytemover software there must be at least one RAM board in the computer beginning at location zero in the memory map. Furthermore, this RAM board must be unprotected for proper execution of the Bytemover software.

Programming Partially Filled PROMs

Software can be loaded into a 2704 or 2708 in as small increments as you desire provided it is added to previously unused areas in that PROM.

This is done by first using Bytemover to move the current contents of the PROM down to RAM, adding the new software to an area of RAM which corresponds to the unused portion of the PROM and finally using Bytemover again to re-program the PROM with the new software.

Although the entire PROM must always be programmed, it never hurts to re-write the same information over again. And, of course, an erased PROM in which all bits are "1" may be programmed at any time.

In general, it is OK to write a "1" over a "1", a "0" over a "0", or a "0" over a "1". But in order to write a "1" over a "0", the PROM must first be completely erased.

If the PROM to which you want to add software is PROM zero on the Bytesaver board, turn off the A.C. power to the computer and install a 1N914 diode just below Q0 (see step 3 of the Bytesaver assembly instructions).

Turn the power back on and move Bytemover down to RAM zero by following Example 1. Add the new software to an area of RAM which corresponds to an unused portion of PROM zero.

Re-program PROM zero by following Example 4 of this manual. Note that you need not erase the PROM to do this.

Turn the computer power off and remove the 1N914 diode below Q0.

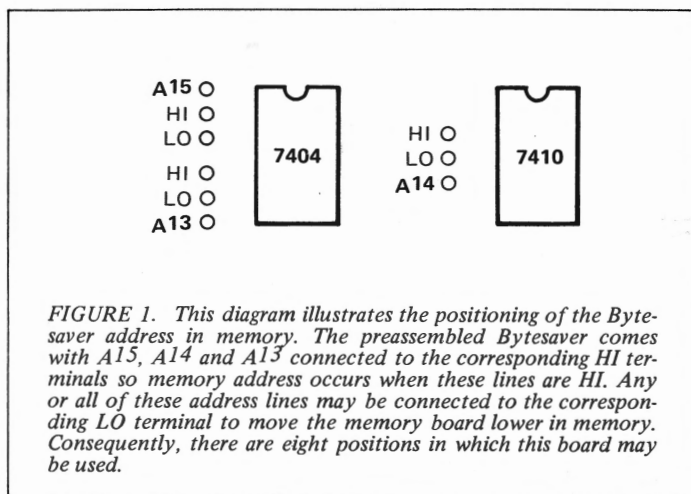
PROM Programming Time

The Bytesaver software supplied here is designed to program a PROM in approximately 30 seconds. We have found that this is generally a sufficiently long period of programming time. However—to be completely within the manufacturer's specifications—the PROM should be programmed for 2 to 3 minutes.

If you wish to program your PROMs for longer than 30 seconds, the Bytemover software may be easily modified. Simply change the contents of location 77H from 40H to 00H. Now you must manually time the programming operation and depress the stop switch at the end of the operation.

Step By Step Instructions

1. Before using the Bytesaver, you must install three jumper wires to set the location of Bytesaver in memory. This adjustment is shown in Figure 1. The assembled Bytesaver comes with A13, A14 and A15 each tied to the corresponding HI pad to position the board at the very top of memory. In the following instructions it is assumed this is the jumper connection used.



2. Turn off all power to the computer and plug in the Bytesaver board.
3. Be sure the program power on the Bytesaver is turned off (program power switch in the down position).
4. Turn on the computer. Raise the reset switch, the stop switch and then raise the reset switch once again to initialize the computer.
5. Raise address switches A15, A14 and A13. All other address switches should be in the down position.
6. Raise the examine switch. You are now examining the contents of the first byte of PROM in PROM location zero of the Bytesaver memory board (memory location 340 000). If the PROM supplied with your Bytesaver is in this PROM location, the data lights will read "061," the first byte of the Bytemover program.

Example 1: Transfer the Bytemover program from PROM to RAM beginning at location zero in RAM.

1. Raise the reset switch.
2. Depress the unprotect switch (on the Altair front panel).
3. Raise A15, A14 and A13. Raise the examine switch. The data lights should read "061" octal.
4. Now set the sense switches for the task to be done, referring to Figure 2.

A15	Down	to transfer from Prom to Ram.
A14	Down	for the transfer of 1K bytes.
A13	Down	All down since we are transferring from the PROM that contains Bytemover (PROM 0).
A12	Down	
A11	Down	
A10	Down	All down for storage to begin at location zero in RAM.
A9	Down	
A8	Down	

A15 	A14 	A13 	A12 	A11 	A10 	A9 	A8 
UP To program a PROM.	UP For a 7K transfer.	MSB PROM address location in increments of 1K from the PROM in which Bytemover is stored.		LSB	MSB Selection of RAM address in 1K increments.		
DOWN To move from PROM to RAM.	DOWN For a 1K transfer.				LSB		

FIGURE 2: Function of the sense switches in Bytemover.

5. Push the run switch. In less than one second, the contents of PROM will be transferred to RAM. The contents of PROM are unaffected by this operation.
6. Raise the stop switch.
7. Raise the reset switch. Note that the data lights read "061".

Example 2: Program a 2708 PROM inserted in PROM location one. This PROM is to be programmed with the contents of the first 1K bytes of RAM beginning at location zero in memory. The Bytesaver software is still in the PROM installed at PROM location zero on the Bytesaver board.

1. Raise the reset switch.
2. Depress the unprotect switch (on the Altair front panel).
3. Raise A15, A14 and A13. Raise the examine switch. The data lights should read "061" octal.
4. Raise the protect switch on the Bytesaver board (i.e. program power switch to the on position). The protect light on the front panel should go off when this switch is raised.
5. Now set the sense switches for the task to be done:

A15	Up	to program a PROM.
A14	Down	(always down for PROM programming).
A13	Down	To select the PROM 1K higher in memory than the PROM that contains Bytemover.
A12	Down	
A11	Up	
A10	Down	All down for transfer to begin at location zero in RAM.
A9	Down	
A8	Down	

6. Push the run switch. Note that panel light A9 is blinking at a rate of about twice per second. When this light stops blinking, the PROM programming is complete.
7. Raise the stop switch.
8. Now note the INTE light on the front panel. If this light is on, the Bytemover Verifier has verified that the contents of the programmed PROM are indeed identical to the contents of the selected 1K bytes of RAM. If this light is off, the PROM has not programmed correctly. This could be due to a defective PROM.

Example 3: Altair 8K BASIC can be stored in seven 2708 PROMs. Given that these seven PROMs are in PROM locations one through seven of the Bytesaver board, 8K BASIC can easily be transferred into RAM using the following procedure:

1. Raise the reset switch.
2. Depress the unprotect switch (on the Altair front panel).
3. Raise A15, A14 and A13. Raise the examine switch. The data lights should read "061" octal.
4. Now set the sense switches for the task to be done:

A15	Down	to transfer from PROM to RAM.
A14	Up	for a 7K transfer.
A13	Down	To begin transfer from the PROM 1K higher in memory than the Bytemover program.
A12	Down	
A11	Up	
A10	Down	All down for storage to begin at location zero in RAM.
A9	Down	
A8	Down	

5. Push the run switch. In less than one second BASIC will be loaded into RAM (it sure beats paper tape!). Now raise the stop switch.

Example 4: If you do not have Bytemover in PROM, you can program a PROM with Bytemover that is stored in RAM. The Bytemover software (see listing) must first be loaded into RAM beginning at location zero in memory. The Bytemover software can then be burned into a PROM using the following procedure:

1. Raise the reset switch.
2. Depress the unprotect switch (on the Altair front panel).
3. Insert an erased PROM into PROM location zero.
4. Examine location 000 240 in memory.
5. Raise the program power switch on the Bytesaver board.
6. Set the sense switches with A15, A14 and A13 up.
7. Push the run switch. When light A9 stops blinking, the programming is complete. The INTE light will be on.
8. Turn off PROM program power by depressing the switch on the Bytesaver.

This is the high order address of prom board

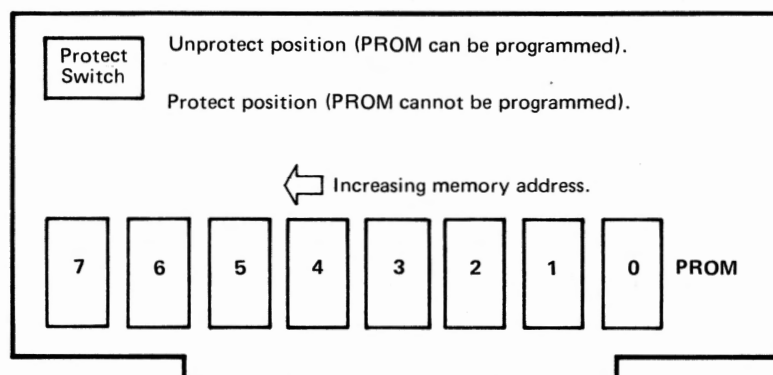


FIGURE 3: Bytesaver physical layout.

Erasing PROMS: The 2704 and 2708 PROMs are erased by shining intense U.V. light through their quartz windows. One such U.V. source is available for \$125 from **Prometrics**, 5345 North Kedzie Av., Chicago, IL 60625.

Bytemover 3.1 Octal Listing

```

061 000 000 301 321 056 311 363 345 345 000 000 000 061 004 000
315 000 000 061 002 000 341 061 004 000 325 305 371 016 000 131
151 333 377 127 346 007 007 007 107 172 346 070 017 000 147 071
056 000 172 353 346 200 017 017 306 055 041 000 000 157 071 351
371 041 013 000 071 353 371 353 021 000 000 073 361 002 003 023
172 346 004 007 007 000 205 157 351 000 000 076 126 205 157 351
000 151 174 140 371 147 056 153 001 000 000 073 361 022 023 003
170 376 374 077 037 037 346 100 056 175 205 157 351 056 153 170
346 004 007 007 007 205 157 351 000 000 000 174 041 000 374 071
371 041 000 374 031 353 147 056 153 170 346 370 306 010 107 351
333 377 107 346 340 036 000 113 127 170 346 037 107 147 056 140
351 306 032 157 333 377 346 100 017 017 205 157 351 174 041 000
374 071 371 056 315 147 351 000 000 000 000 373 351 174 041 000
374 031 353 056 361 147 001 000 000 351 000 326 220 157 172 306
004 127 376 070 077 076 000 037 205 157 351 000 000 373 351 351
351 073 361 353 276 353 027 346 001 057 074 205 157 073 073 361
057 353 206 353 306 007 077 027 346 001 057 074 205 157 003 023
170 346 004 057 074 205 157 351 000 000 000 000 000 000 000 000
    
```

BYTEMOVER ASSEMBLY LISTING

Cromemco

0000		0000 * BYTEMOVER (T.M.) SOFTWARE FOR
0000		0001 * CROMEMCO 8K BYTESAVER (T.M.)
0000		0002 * VERSION 3.1
0000		0003 * SELF-RELOCATING SOFTWARE LOCATABLE AT ANY
0000		0004 * 1024 BYTE (1K) BOUNDARY IN MEMORY
0000		0009 * ROUTINE TO FIND ONESELF IN MEMORY
0000		0010 SP EQU 6
0000		0019 * DEFINE FIRST 4 BYTES IN MEMORY AS STACK
0000	31 00 00	0020 LXI SP, 0
0003		0029 * SAVE FIRST FOUR BYTES IN REGISTERS
0003	C1	0030 POP B
0004	D1	0040 POP D
0005		0049 * REPLACE BYTE 0 WITH A 'RETURN'
0005	2E C9	0050 MVI L, 0C9H
0007	F3	0051 DI
0008	E5	0060 PUSH H
0009	E5	0070 PUSH H
000A	00	0080 NOP
000B	00	0081 NOP
000C	00	0082 NOP
000D	31 04 00	0090 LXI SP, 4
0010	CD 00 00	0100 CALL 0
0013		0101 * ROM LOCATION NOW IN BYTE 3
0013	31 02 00	0110 LXI SP, 2
0016	E1	0120 POP H
0017		0129 * RETURN BYTES 0-3
0017	31 04 00	0130 LXI SP, 4
001A	D5	0140 PUSH D
001B	C5	0150 PUSH B
001C		0159 * STORE ROM LOCATION IN SP
001C	F9	0160 SPHL
001D	0E 00	0170 MVI C, 0
001F	59	0180 MOV E, C
0020	69	0190 MOV L, C
0021		0199 * INPUT SENSE SW COMMANDS
0021	DB FF	0200 IN 255
0023	57	0210 MOV D, A
0024		0219 * STRIP RAM ADDRESS
0024	E6 07	0220 ANI 7
0026	07	0230 RLC
0027	07	0240 RLC
0028		0249 * STORE RAM ADDRESS IN BC
0028	47	0250 MOV B, A
0029	7A	0260 MOV A, D
002A		0269 * STRIP ROM ADDRESS
002A	E6 38	0270 ANI 56
002C	0F	0280 RRC
002D	00	0290 NOP
002E	67	0300 MOV H, A
002F	39	0310 DAD SP
0030	2E 00	0320 MVI L, 0
0032	7A	0330 MOV A, D
0033	EB	0340 XCHG
0034		0341 * ADDRESS OF ROM BEING PROCESSED IN DE
0034		0349 * BRANCH TO TRANSFER OR PROGRAM ROUTINE

BYTEMOVER ASSEMBLY LISTING**Cromemco**

0034 E6 80	0350 ANI 128
0036 0F	0360 RRC
0037 0F	0370 RRC
0038 C6 2D	0380 ADI 45
003A 21 00 00	0390 LXI H, 0
003D 6F	0400 MOV L, A
003E 39	0410 DAD SP
003F E9	0420 PCHL
0040	0500 * ROUTINE TO TRANSFER ROM TO RAM
0040 F9	0510 SPHL
0041 21 0B 00	0520 LXI H, 11
0044 39	0530 DAD SP
0045 EB	0550 XCHG
0046 F9	0560 SPHL STACK CONTAINS ROM LOCATION
0047 EB	0570 XCHG H&L CONTAIN LOOP ADDRESS
0048 11 00 00	0580 LXI D, 0
004B	0588 * START OF TRANSFER LOOP
004B	0589 * INCREMENT ROM ADDRESS
004B 3B	0590 DCX SP
004C	0599 * MOVE DATA FROM ROM TO RAM
004C F1	0600 POP 6
004D 02	0610 STAX B
004E	0619 * INCREMENT RAM ADDRESS
004E 03	0620 INX B
004F	0629 * INCREMENT BYTE COUNT
004F 13	0630 INX D
0050 7A	0640 MOV A, D
0051 E6 04	0650 ANI 4
0053 07	0660 RLC
0054 07	0670 RLC
0055 00	0680 NOP
0056 85	0690 ADD L
0057 6F	0700 MOV L, A
0058 E9	0710 PCHL
0059 00	0716 NOP
005A 00	0717 NOP
005B	0719 * JUMP TO 00B1 FROM TRANSFER ROUTINE
005B 3E 56	0720 MVI A, 56H
005D 85	0725 ADD L
005E 6F	0730 MOV L, A
005F E9	0740 PCHL
0060	1000 * ROUTINE TO PROGRAM ROM
0060 00	1010 NOP
0061	1019 * MOVE RAM ADDRESS INTO HL
0061 69	1020 MOV L, C
0062 7C	1030 MOV A, H
0063 60	1040 MOV H, B
0064	1049 * MOVE RAM ADDRESS INTO SP
0064 F9	1050 SPHL
0065 67	1060 MOV H, A
0066 2E 6B	1070 MVI L, 107
0068	1079 * INCREMENT RAM ADDRESS
0068 01 00 00	1080 LXI B, 0
006B	1089 * INCREMENT RAM ADDRESS
006B 3B	1090 DCX SP
006C	1098 * USE STAX AND POP 6 (PSW)

BYTE MOVER ASSEMBLY LISTING**Cromemco**

006C		1099	* TO MOVE DATA FROM ROM TO RAM
006C F1		1100	POP 6
006D 12		1110	STAX D
006E		1119	* INCREMENT ROM ADDRESS
006E 13		1120	INX D
006F		1129	* INCREMENT BYTE COUNT
006F 03		1130	INX B
0070		1138	* B STORES TWO CONSTANTS
0070		1139	* # COMPLETE PASSES & IN ROM CNT
0070 78		1140	MOV A, B
0071		1149	* # PASSES = 32 ?
0071 FE FC		1150	CPI 252
0073 3F		1160	CMC
0074 1F		1170	RAR
0075 1F		1180	RAR
0076		1198	* SET 64 TO 0 FOR TWO MINUTE TIMER VERSION
0076 E6 40		1200	ANI 64
0078		1201	* A=64 IF COMPLETED 32 PASSES
0078 2E 7D		1205	MVI L, 7DH
007A 85		1210	ADD L
007B 6F		1220	MOV L, A
007C E9		1225	PCHL
007D 2E 6B		1226	MVI L, 6BH
007F 78		1230	MOV A, B
0080 E6 04		1240	ANI 4
0082		1241	* A=4 IF END OF 1024 BYTE PASS
0082 07		1250	RLC
0083 07		1260	RLC
0084 07		1270	RLC
0085 85		1280	ADD L
0086 6F		1290	MOV L, A
0087		1291	* GO BACK TO 1090 UNLESS OVERFLOW
0087		1292	* THEN GO TO 1380 FOR
0087		1293	* ADDRESS SUBTRACTION
0087		1294	* OR 2135 FOR QUIT
0087 E9		1300	PCHL
0088 00		1350	NOP
0089 00		1360	NOP
008A 00		1370	NOP
008B		1378	* ANOTHER PROGRAM PASS TO BE DONE
008B		1379	* ADJUST ROM AND RAM ADDRESSES
008B 7C		1380	MOV A, H
008C 21 00 FC		1390	LXI H, 64512
008F		1399	* SUBTRACT 1024 FROM ROM ADDRESS
008F 39		1400	DAD SP
0090 F9		1410	SPHL
0091 21 00 FC		1420	LXI H, 64512
0094		1429	* SUBTRACT 1024 FROM RAM ADDRESS
0094 19		1430	DAD D
0095 EB		1440	XCHG
0096 67		1450	MOV H, A
0097 2E 6B		1460	MVI L, 107
0099 78		1470	MOV A, B
009A E6 FB		1480	ANI 248
009C		1489	* INCREMENT PASS COUNTER BY ONE
009C C6 08		1490	ADI 8

BYTEMOVER ASSEMBLY LISTING *Cromemco*

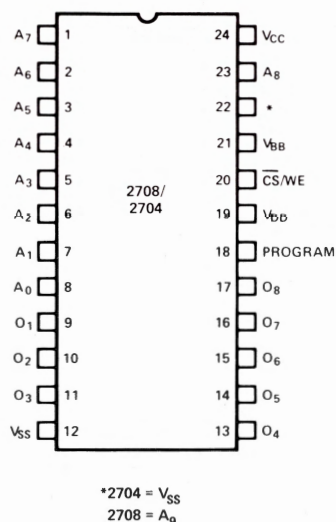
009E 47	1495 MOV B, A
009F	1499 * GO BACK TO 1090
009F E9	1500 PCHL
00A0	2000 * ROUTINE TO LOAD BYEMOVER INTO ROM
00A0 DB FF	2010 IN 255
00A2 47	2020 MOV B, A
00A3 E6 E0	2030 ANI 224
00A5 1E 00	2040 MVI E, 0
00A7 4B	2050 MOV C, E
00A8 57	2060 MOV D, A
00A9 7B	2070 MOV A, B
00AA E6 1F	2080 ANI 31
00AC 47	2090 MOV B, A
00AD 67	2100 MOV H, A
00AE 2E 60	2110 MVI L, 96
00B0 E9	2120 PCHL
00B1	2121 * CHECK FOR 7K TRANSFER OF ROM TO RAM
00B1 C6 1A	2122 ADI 1AH
00B3 6F	2123 MOV L, A
00B4 DB FF	2124 IN 255
00B6 E6 40	2125 ANI 64
00B8 0F	2126 RRC
00B9 0F	2127 RRC
00BA 85	2128 ADD L
00BB 6F	2129 MOV L, A
00BC E9	2130 PCHL
00BD	2133 * PROGRAMMER VERIFICATION ROUTINE
00BD	2134 * PART 1
00BD 7C	2135 MOV A, H
00BE 21 00 FC	2145 LXI H, 64512
00C1 39	2155 DAD SP
00C2 F9	2165 SPHL
00C3 2E CD	2175 MVI L, OCDH
00C5 67	2185 MOV H, A
00C6 E9	2195 PCHL
00C7 00	2205 NOP
00C8 00	2210 NOP
00C9 00	2215 NOP
00CA 00	2220 NOP
00CB	2229 * ROM TO RAM TRANSFER STOP ROUTINE
00CB FB	2230 EI
00CC E9	2240 PCHL
00CD	2248 * PROGRAMMER VERIFICATION ROUTINE
00CD	2249 * PART 2
00CD 7C	2250 MOV A, H
00CE 21 00 FC	2260 LXI H, 64512
00D1 19	2270 DAD D
00D2 EB	2280 XCHG
00D3 2E F1	2290 MVI L, 0F1H
00D5 67	2300 MOV H, A
00D6 01 00 00	2310 LXI B, 0
00D9 E9	2320 PCHL
00DA 00	2625 NOP
00DB	2629 * 7K TRANSFER COMPLETION CHECK
00DB D6 90	2630 SUI 90H
00DD 6F	2640 MOV L, A

BYTEMOVER ASSEMBLY LISTING

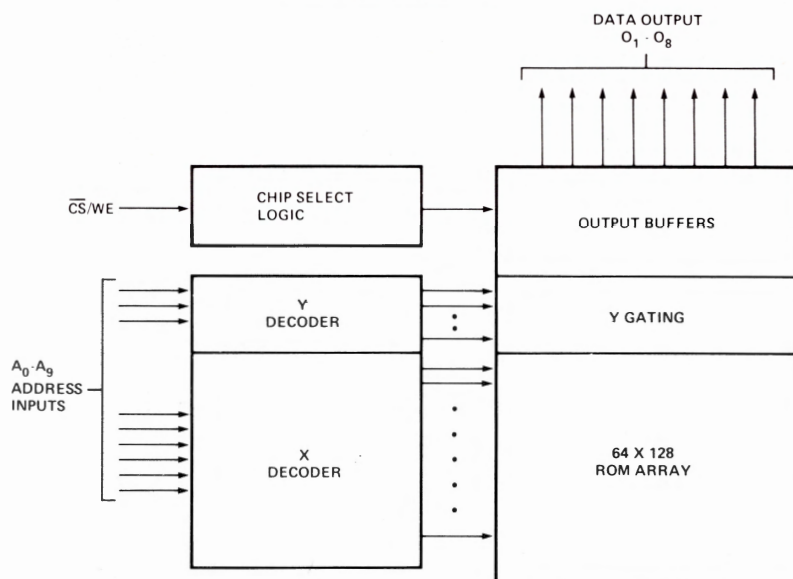
Cromemco

00DE 7A	2650 MOV A, D
00DF C6 04	2660 ADI 4
00E1 57	2670 MOV D, A
00E2 FE 38	2680 CPI 56
00E4 3F	2685 CMC
00E5 3E 00	2690 MVI A, 0
00E7 1F	2700 RAR
00E8 85	2710 ADD L
00E9 6F	2720 MOV L, A
00EA E9	2730 PCHL
00EB	2879 * ROM PROGRAMMER STOP ROUTINE
00EB 00	2880 NOP
00EC 00	2881 NOP
00ED FB	2885 EI
00EE E9	2890 PCHL
00EF E9	2900 PCHL
00F0 E9	2906 PCHL
00F1	2918 * PROGRAMMER VERIFICATION ROUTINE
00F1	2919 * PART 3
00F1 3B	2920 DCX SP
00F2 F1	2930 POP 6
00F3 EB	2940 XCHG
00F4	2949 * COMPARE FOR GREATER
00F4 BE	2950 CMP M
00F5 EB	2960 XCHG
00F6 17	2970 RAL
00F7 E6 01	3000 ANI 1
00F9 2F	3010 CMA
00FA 3C	3011 INR A
00FB 85	3015 ADD L
00FC 6F	3020 MOV L, A
00FD 3B	3030 DCX SP
00FE 3B	3040 DCX SP
00FF	3050 * COMPARE FOR LESSER
00FF F1	3055 POP 6
0100 2F	3056 CMA
0101 EB	3058 XCHG
0102 86	3059 ADD M
0103 EB	3060 XCHG
0104 C6 07	3061 ADI A, 1
0106 3F	3065 CMC
0107 17	3070 RAL
0108 E6 01	3090 ANI 1
010A 2F	3100 CMA
010B 3C	3101 INR A
010C 85	3105 ADD L
010D 6F	3110 MOV L, A
010E 03	3130 INX B
010F 13	3140 INX D
0110 78	3150 MOV A, B
0111 E6 04	3180 ANI 4
0113 2F	3190 CMA
0114 3C	3191 INR A
0115 85	3195 ADD L
0116 6F	3200 MOV L, A
0117 E9	3210 PCHL

PIN CONFIGURATIONS



BLOCK DIAGRAM



PIN NAMES

A ₀ -A ₉	ADDRESS INPUTS
O ₁ -O ₈	DATA OUTPUTS
CS/WE	CHIP SELECT/WRITE ENABLE INPUT

READ OPERATION

DC & Operating Characteristics

T_A = 0°C to 70°C, V_{CC} = +5V ±5%, V_{DD} = +12V ±5%, V_{BB} = -5V ±5%, V_{SS} = 0V, Unless Otherwise Noted.

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Conditions
I _{LI}	Address and Chip Select Input Load Current			10	μA	V _{IN} = 5.25V
I _{LO}	Output Leakage Current			10	μA	V _{OUT} = 5.25V, CS/WE = 5V
I _{DD}	V _{DD} Supply Current		50	65	mA	Worst Case Supply Currents: All Inputs High CS/WE = 5V; T _A = 0°C
I _{CC}	V _{CC} Supply Current		6	10	mA	
I _{BB}	V _{BB} Supply Current		30	45	mA	
V _{IL}	Input Low Voltage	V _{SS}		0.65	V	
V _{IH}	Input High Voltage	3.0		V _{CC} +1	V	
V _{OL}	Output Low Voltage			0.45	V	I _{OL} = 1.6mA
V _{OH1}	Output High Voltage	3.7			V	I _{OH} = -100μA
V _{OH2}	Output High Voltage	2.4			V	I _{OH} = -1mA
P _D	Power Dissipation			800	mW	T _A = 70°C

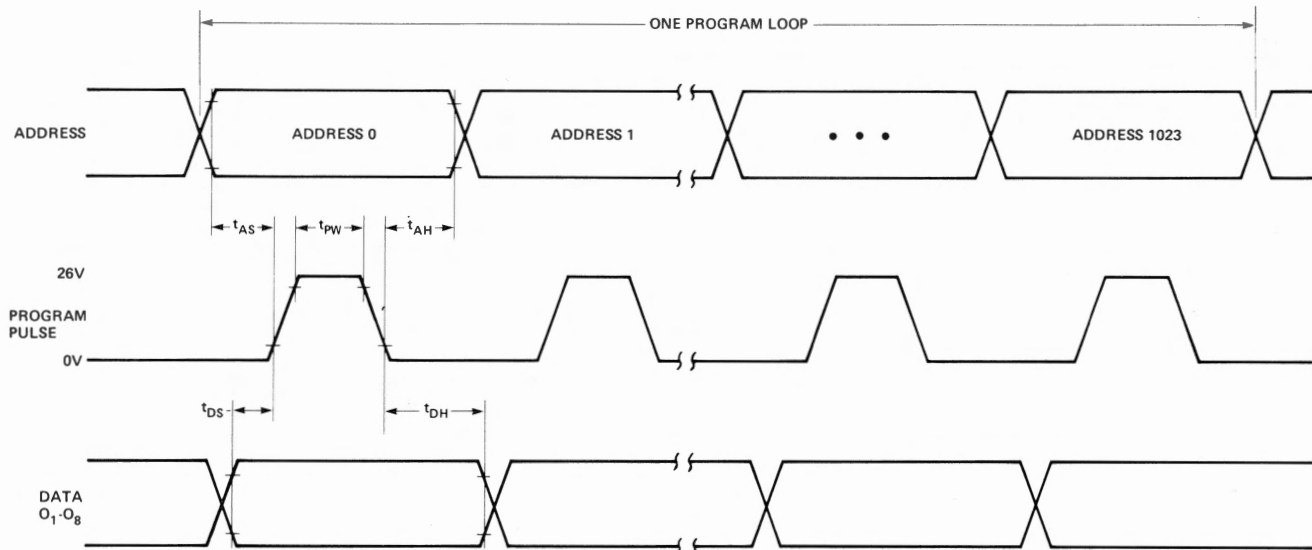
NOTES: 1. Typical values are for T_A = 25°C and nominal supply voltages.
2. The program input (Pin 18) may be tied to V_{SS} or V_{CC} during the read mode.

Waveforms

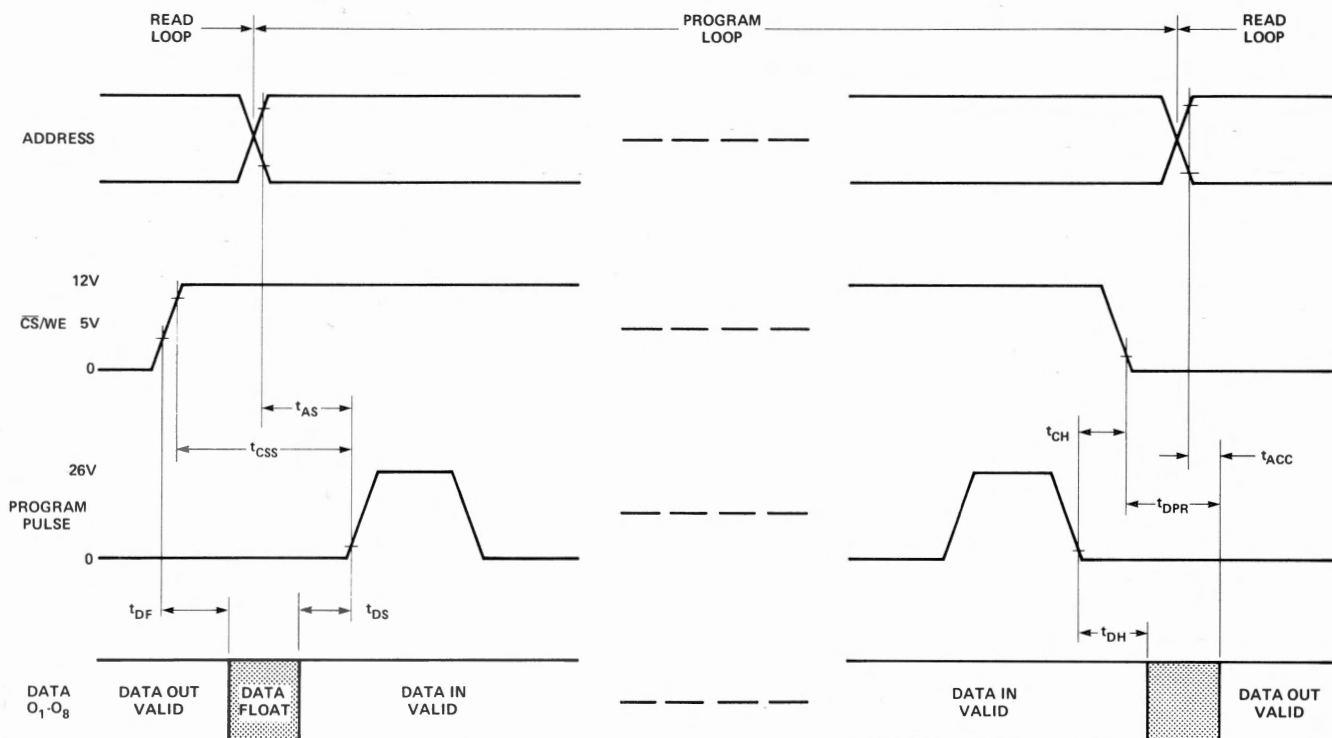
(Logic levels and timing reference levels same as in the Read Mode unless noted otherwise.)

Program Mode

$\overline{CS}/WE = +12V$



Read/Program/Read Transitions



AC Characteristics

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = +5\text{V} \pm 5\%$, $V_{DD} = +12\text{V} \pm 5\%$, $V_{BB} = -5\text{V} \pm 5\%$, $V_{SS} = 0\text{V}$, Unless Otherwise Noted.

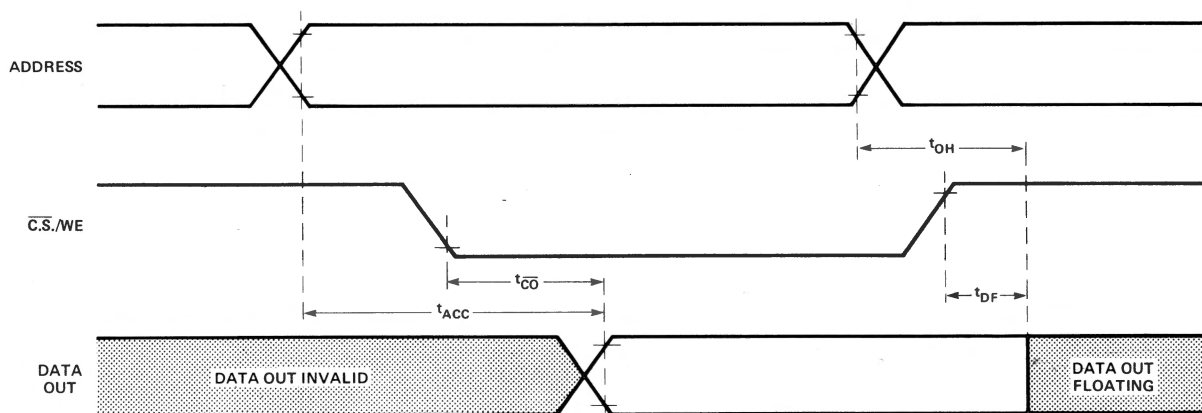
Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{ACC}	Address to Output Delay		280	450	ns
t_{CO}	Chip Select to Output Delay			120	ns
t_{DF}	Chip De-Select to Output Float	0		120	ns
t_{OH}	Address to Output Hold	0			ns

Capacitance^[1] $T_A = 25^\circ\text{C}$, $f = 1\text{MHz}$

Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN}	Input Capacitance	4	6	pF	$V_{IN}=0\text{V}$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT}=0\text{V}$

Note 1. This parameter is periodically sampled and not 100% tested.

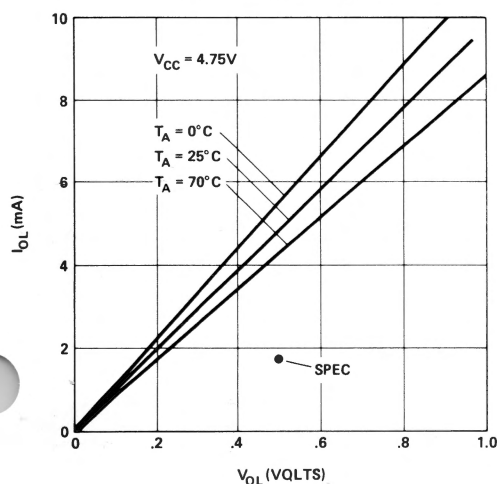
Waveforms



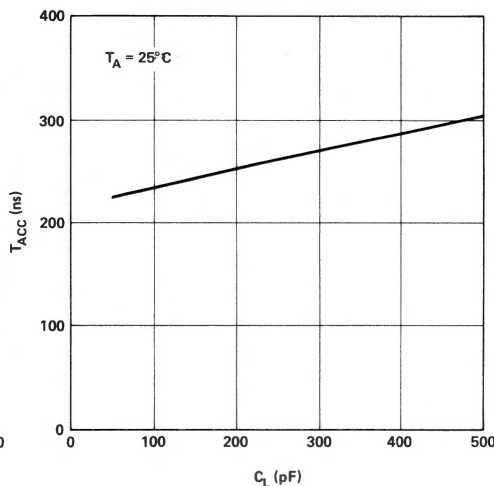
Typical Characteristics

(Nominal supply voltages unless otherwise noted):

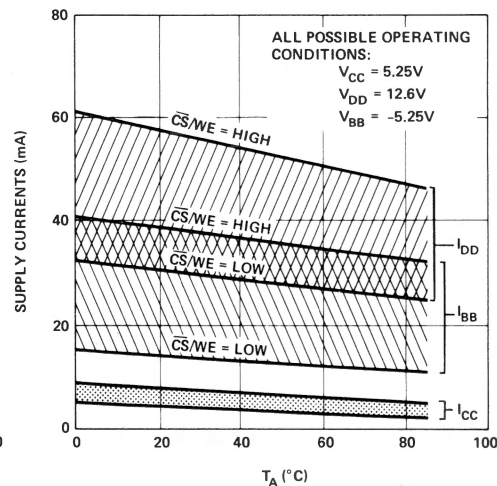
OUTPUT SINK CURRENT
VS. OUTPUT VOLTAGE



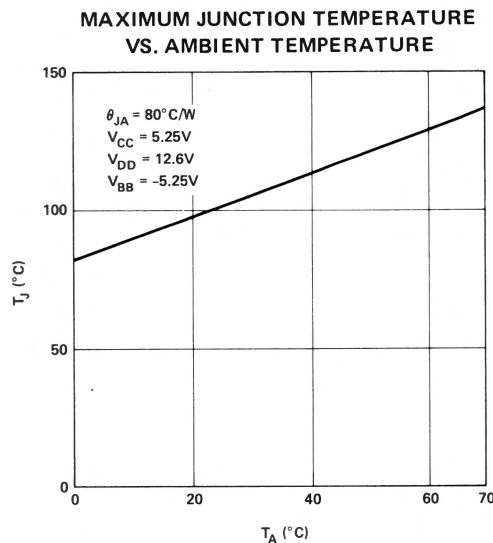
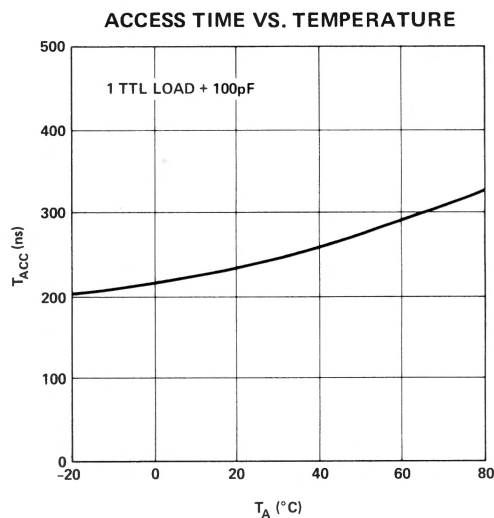
ACCESS TIME
VS. LOAD CAPACITANCE



RANGE OF SUPPLY CURRENTS
VS. TEMPERATURE



2708-2704 PROM DATA



PROGRAMMING OPERATION

Description

Initially, and after each erasure, all bits of the 2708/2704 are in the "1" state (Output High). Information is introduced by selectively programming "0" into the desired bit locations.

The circuit is set up for programming operation by raising the $\overline{\text{CS/WE}}$ input (Pin 20) to +12V. The word address is selected in the same manner as in the read mode. Data to be programmed are presented, 8-bits in parallel, to the data output lines (O_1 - O_8). Logic levels for address and data lines and the supply voltages are the same as for the read mode. After address and data set up one program pulse (V_P) per address is applied to the program input (Pin 18). One pass through all addresses to be programmed is defined as a program loop. The number of loops (N) required is a function of the program pulse width (t_{PW}) according to $N \times t_{PW} \geq 100 \text{ ms}$.

For program verification, program loops may be alternated as shown on page 12.

Program Characteristics

$T_A = 25^{\circ}\text{C}$, $V_{CC} = +5\text{V} \pm 5\%$, $V_{DD} = +12\text{V} \pm 5\%$, $V_{BB} = -5\text{V} \pm 5\%$, $V_{SS} = 0\text{V}$, $\overline{\text{CS/WE}} = +12\text{V}$, Unless Otherwise Noted.

Symbol	Parameter	Min.	Typ.	Max.	Units
t_{AS}	Address Setup Time	10			μs
t_{CSS}	$\overline{\text{CS/WE}}$ Setup Time	10			μs
t_{DS}	Data Setup Time	10			μs
t_{AH}	Address Hold Time	1			μs
t_{CH}	$\overline{\text{CS/WE}}$ Hold Time	.5			μs
t_{DH}	Data Hold Time	1			μs
t_{DF}	Chip Deselect to Output Float Delay	0		120	ns
t_{DPR}	Program To Read Delay			10	μs
t_{PW}	Program Pulse Width	.1		1.0	ms
t_{PR}	Program Pulse Rise Time	.5		2.0	μs
t_{PF}	Program Pulse Fall Time	.5		2.0	μs
I_P	Programming Current		10	20	mA
V_P	Program Pulse Amplitude	25		27	V